

How To Make A WebToon

Pamela P. Walatka

NAS-98-014

December 1998

*walatka@nas.nasa.gov
MRJ Technology Solutions
NASA Ames Research Center
Mail Stop T27A-2
Moffett Field, CA 94035-1000*

Abstract

A WebToon is a cartoon-style table of contents for HTML documents on the web, using imagemaps that are pictures of people talking. The people have word balloons above their heads; the word balloons serve as links to relevant pages. A WebToon adds the human touch to web interfaces, and makes web pages more approachable, interesting, and easier to navigate.

Basically, to make a WebToon, you create a one-page collection of imagemaps featuring people talking. The talk is in word balloons that *link* to a relevant part of a document. The imagemaps are combined in a table that fits on one page.

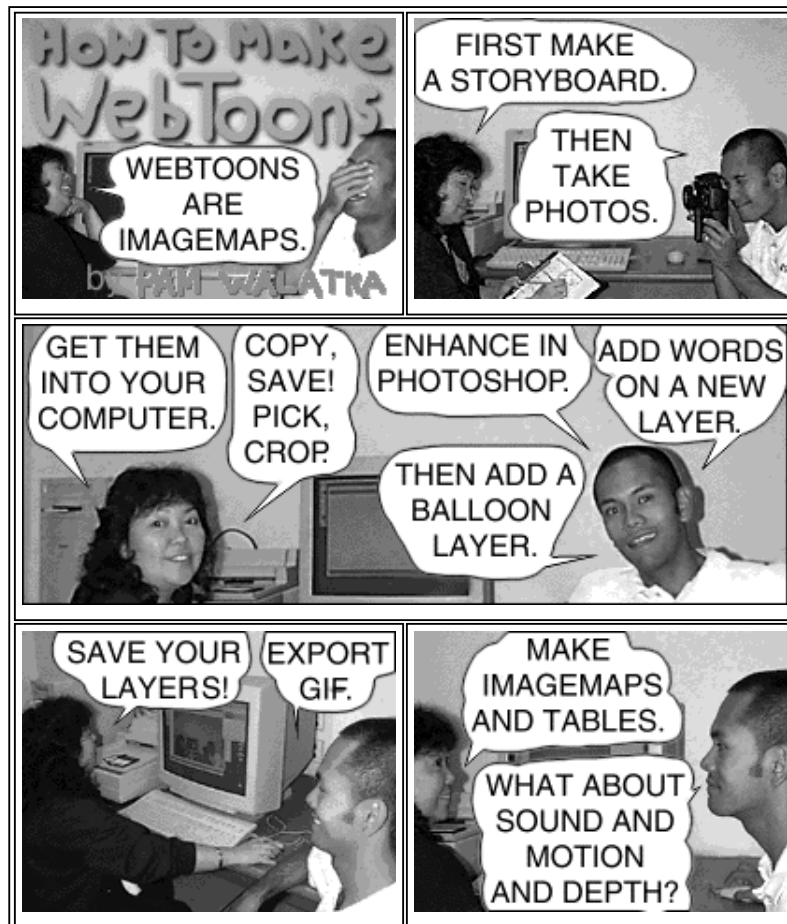
First organize your long document into chapters. Sketch a storyboard with one panel (image) for each chapter. Take photos of people acting out your storyboard. Use image-processing software such as Adobe Photoshop to crop and enhance your photos. Use a new image layer to add words that briefly describe each chapter. Use another layer to put white balloons behind the words. Make each image into a client-side imagemap. Organize the imagemaps with an HTML table.



This report explains the details of how to make a WebToon.

Table of Contents

The 'Toon About the 'Toons



In the on-line version of this report, the WebToons link to parts of the report. The user clicks on a word balloon and is taken to the relevant section.

Chapter One

WebToons Are Imagemaps

Introduction

The purpose of WebToons is to humanize and organize the interfaces to long documents. WebToons enhance the usability of the web interface. WebToon imagemaps are like normal imagemaps (which are explained below; what is different is the use of pictures of people, and the cartoon-style word balloons. The people and the cartoon style make the interface more approachable.

The creation of a WebToon forces you to organize your long document, in order to be able to condense it into a few logical words that fit into word balloons.

In a WebToon, each panel is a separate imagemap. Working with separate imagemaps gives you the option of making quick corrections to a small image and the flexibility to rearrange the images. Alternately, it could be argued that working with one composite of the images is easier. This point will be addressed at the end of the Photoshop section.

Basic Imagemap

Imagemaps are images that contain multiple links. The user clicks somewhere in the image, and a new resource is retrieved: it could be another image, another page, a sound or anything else that can be represented by a URL. Just as a normal image can link to a single URL, an imagemap links to multiple URLs. You, the page creator, must map out the areas of the image that you want to relate to each specific link.

Here is a client-side imagemap. See the online version at <http://science.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-98-014/chapter1.html> to see how it works.



In client-side imagemaps, the list of map coordinates is embedded in the HTML document. For details, see Chapter 5.

Chapter Two

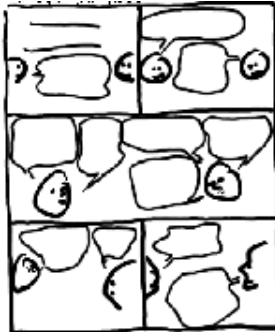
Storyboards and Photos

First Make a Storyboard



A storyboard is a **rough** sketch of how the WebToon is going to look. I use paper and pencil. Points to keep in mind:

- It all has to fit on one page, because, by definition, WebToons condense documents into just a few sentences. A central idea of the WebToons is that they force the document author to present ideas in brief logical bits.
- First sketch out some boxes (panels). Generally nine boxes are the maximum that fit on one page. Five is good, with the middle one long.



See Appendix A for sample storyboard layouts you can use.

- Rough in some word balloons.
- Start filling in the word balloons with the major points of your document. This is **very** hard, and may require several iterations. Get a new sheet of paper and start over. Or you could make copies of the boxes and balloons

and try different ways of filling in the words. If you are the author of the document, you might find yourself reorganizing the chapters to make more sense. Each panel should represent one chapter or section. The words have to fit in the balloons; you **MUST** condense. Take some time; do it many times until you have the essence.

- If you have an idea of what the actors should be doing, sketch them in, without worrying about how good they look. For example, when I was sketching frame two of these Toons, I knew I needed Joel to be using a camera.
- You are finished with the storyboard when you have a set of simple, easy-to-understand word balloons that show the key ideas of your document, along with the positions of your actors.

Creating a WebToon Image

The next phase involves creating the WebToon digital images based on the storyboard. Now you have a style choice to make: do you want photographic images or traditional-cartoon drawings?

You could draw the images

There is no requirement for the images used in WebToons to be photographic. I originally tried drawing them in Adobe Illustrator, but the results were poor, due to my drawing ability. One of the hardest parts was getting the characters to look consistent from panel to panel. If you can draw, I encourage you to try it, electronically or on paper (and then scan). Eventually, the images must be digital in order to be made into imagemaps on the web.

Photos are fast

I found photography a quick and easy means of getting images of people.

Taking Photographs for WebToons

Before you start taking pictures, make sure you have a **COMPLETED** storyboard and you therefore know what you want your actors to do. This will expedite the filming process. For actors, use your friends and colleagues. Give your actors at least a day's warning so that they can prepare what to wear. Perhaps solid color clothing would be easier to "select" when you are working with the images, later. Use a setting with plenty of light.

- **Digital camera is fastest**

The fastest and most technically elegant way to get the digital images for WebToons is to use a digital camera of good quality. You will need a resolution of at least 400 x 300 pixels; this gives you the chance to crop down to 200 x 150. Another alternative would be to use a regular film

camera and then scan in the photographs. My final images are 200 x 150 at 96 dpi.

Note: be sure to test your camera, in the room that you will be using, and download a few test images to make sure that the camera settings are correct.

- **Leave room over the actors' heads**

This is easy to forget. When you are taking the photos, imagine the word balloons over the heads. The general rule of thumb is to leave the top HALF of the image for the balloons.

- **Take twice as many snapshots as you think you need**

More is better. You can choose which ones to use later. You may have to take many shots before the actors relax. Sometimes the best shots will surprise you.

- **Use props**

Whenever possible, have something in the picture to illustrate what you are talking about. If you are talking about fire extinguishers, show the people holding up a fire extinguisher. (Later, when you make imagemaps, have significant parts of the fire extinguisher be links to relevant information.) If you are talking about something with visible-physical steps, show the steps in your images.

Chapter Three

Working With the Images

Get the images into your computer

- **From digital camera**

Many digital cameras come with an A/V line that you hook up to a computer that has the appropriate software for loading the images. Other cameras write the photos to a removable disk.

- **From film**

You can have your film developed onto a CD ROM, or get prints and then scan at medium resolution. You want to have at least 400 pixels wide by 300 high; this gives you room to crop.

- **From drawings**

If you drew the images on paper, scan them. If you drew them electronically, they are already in your computer.

Image Processing

At this point, you are ready to start working with the images in your computer. First you will make backups, then load your files into an image processing program, pick out which images to use, and crop them to get just the important parts of the images. Then you will create layers: background, people, balloons, and words. You will save your layered files for future revisions, and save out each panel as a non-layered GIF file.

Photoshop from Adobe works well as an image processing program for WebToons. The following instructions include details about using Photoshop. NAS staff and users: Photoshop is available on the NAS SGIs; just type `photoshop` at the Irix prompt. The latest version for the SGI is 3.0.1, which these instructions are based on. A later version of Photoshop is also available on the machines in the NAS Multimedia Lab.

Pick and crop the images you want to use


- **Make extra copies**

You may mess up your images and want to go back to the originals, or need to copy a part of the image out of the original. Make and save copies of all your originals.

- **Look, list, pick**

Load your image files into an image processing program such as Photoshop. Spend some time looking at the images you have. Bring as many as possible up on your screen at the same time. Give each one a representative name, and make a list of what you have. Pick the most expressive and interesting shots, and decide which will go where. That is, pick one image for each panel of your storyboard. (A cartoon panel is one frame or box or picture-- nine or fewer panels make one page or one WebToon).

- **Crop, leaving extra space on top**

Use your image processing program to crop all the images to the same (or coordinated) size(s). In this WebToon, I used 200 x 150 pixels at 96 dpi for each panel, except for the extra-wide middle panel, which is 400 x 150. The size will depend on your storyboard; remember that you want the entire WebToon to fit on one browser screen. In Photoshop, you can set the cropping tool to repeat the same dimensions and resolution. Double-click on the Crop tool  and the Cropping Tools Options dialog box will appear. Click Fixed Target Size. Crop down to the most interesting parts of the image. Remember to leave room--the top half of the image--for the word balloons.

Add layers to your images, and enhance

If your image processing program offers layers, you can create several different layers of your image, and work separately on each layer. Working on layers allows you to make changes to one layer without destroying the pixels on the other layers. It would be possible but treacherous to make WebToons without layers, because without layers each one of your changes would leave permanent changes on the whole image.

- **Photoshop (3 or above) offers layers**

Layers make it possible, for example, to rewrite the words without changing the pixels on the other parts of the image. This is important; you may need to write the words many times before you get them to fit your available space. When you are working with layers, you put the words on a separate layer and the balloons on a separate layer to facilitate adjustments and corrections.

The first thing to do is make a copy of the background layer. In Photoshop, select Window/Palettes/Show Layers to get the Layers palette. Drag the rectangle representing the Background layer down to the new-layer icon.



You now have two layers. The top one is titled "Background copy," the bottom one "Background." Next, click the eye of the Background layer

off, and never look at or work on the background again, unless you need to replace messed-up pixels. The active layer is highlighted in the layer palette--click on a layer name to highlight it and make it the layer you are working on. (When you get a message like "no-pixels-selected," you probably are not working on the layer you think you are.)

By the time you are done, your image should have the following layers (from the top down):


1. Words
2. Balloons
3. People
4. Background copy
5. Background

See the snapshot of a Photoshop session at the top of Chapter Four to see how the layer palette looks.

Remember that each panel is a separate image which will become a separate imagemap which you will assemble into a table with borders. You do not need to make borders on your images; HTML will do it for you, through the TABLE tag.

● Use layers to pop the colors (people)

When an artist draws cartoons, the artist can make the characters stand out from the background by making the characters bigger than normal, and more vividly colored. With photographs, there are ways to make the characters stand out. One way is called "popping the color." To pop the color means to isolate the subject of interest and then make the rest of the image grayscale. To do that in Photoshop:


1. Drag the Background copy layer to the new layer icon. Name the new layer People.
2. Select the characters
 - Check to see that the People layer is highlighted in the Layers palette, to indicate that it is the active layer.
 - Make a very rough selection of one of the people by choosing the lasso tool , holding down Alt or Option, and clicking around the person.
 - Enter Mask mode: make the foreground color black by clicking


on the tiny black box near the color indicator boxes



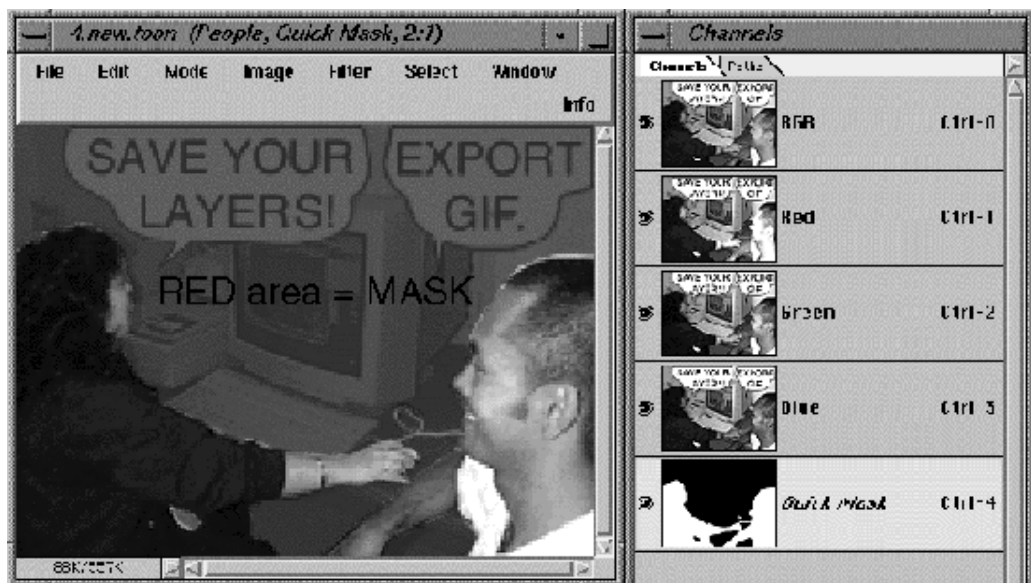
. Go into mask mode by clicking on the right-hand mask icon.

- Refine your selection in Mask mode: use the paintbrush. Click

on the paintbrush  and paint away any areas of the background that are not under the red mask. Use the Brushes palette to change the size of your brush as necessary. If parts of the person are covered by the mask, click on the tiny arrows near the color indicator boxes to change the current color to white. (Also click the tiny boxes if the background color was not white.) Using white removes the mask. Keep working until you have a pretty good selection of the person. Remember, brushing with black adds to the mask; brushing with white removes the mask. The areas not under the mask are the areas that will become selected as you exit mask mode. That is, mask mode enables you to brush in the details of any selection you are trying to make. You can adjust the edge of your selection back and forth until you get it right.

- Exit Mask mode: when you click on the left-side quick mask icon  the mask goes away and your unmasked areas are selected. That is, you have used Mask mode to create a precise selection.



- Then choose Select/Save Selection (the default #4 is fine). You now have a reusable selection of your person or character.



Review of Photoshop Quick Mask. The red area is the mask. The not-red areas become the selection when you exit Quick Mask mode. To add to the mask area, paint with black. To add to the selection, paint with white. Note that the palette showing on the right is the Channels palette (not Layers). You choose which palette to show, from the Windows menu.

3. Repeat the mask-selection process with any remaining person or persons or characters in that image.

4. When you have finished selecting the areas (characters) that you want to "pop" in the first image, retrieve the first selection you saved (channel 4) by choosing Select/Load Selection/OK. If you saved more than one mask-selection, repeat Load Selection, changing the channel number on the Load Selection dialog box to the next one up, and clicking Add to Selection. Repeat until you have loaded all the selections you saved for this image.
5. All your people should now be selected. Choose Select/Save Selection to save the combined selections for possible future use.
6. Choose Select/Inverse to select the area around the people.
7. Hit the Delete or Backspace key to remove the background from this layer. You will not see any difference, because the background is still on the layer underneath. To verify that the background was removed from the people layer, click on the eye by the Background Copy layer in the layers palette to make the background layer invisible. Then click the eye again.
8. In the layers palette, click on the Background Copy layer to make it active. Choose Image/Adjust/Desaturate, which will turn your colored background to gray. Your people should now be popped.

Also, you can now add a halo around the people. On the Layers palette, click the new layer  icon, and then, still in the Layers palette, drag the new layer so that it is between the People layer and Background Copy layer. Select the paintbrush  icon, and choose a fuzzy brush from the Brushes palette, and white paint as the foreground color. With the new layer active (highlighted), just brush at the edge of the people on the image.

● **Adjust levels; enhance as necessary**

With the People layer active, choose Image/Adjust/Levels. In the middle of the Levels dialog box, under Input Levels, you will see a histogram (graph) representing the brightness and darkness levels in your people layer. Drag the little black triangle on the left about half an inch toward the middle. Click OK. Do the same with the white triangle on the right--drag toward the middle. Adjusting the levels like this makes your grayish tones more black and white, increasing punch.

Depending on your available time and talent, make other enhancements such as using the Photoshop filters to create effects, or using the Photoshop Image/Adjust options to give the colors more punch.


At this point, I would recommend that you go back and bring your other images up to this stage before adding the words and balloons. Be sure that all your images receive the same treatments; there should be a consistent look to your images. Remember that you should have at least three and not more than nine images, one for each panel of the WebToon, and that the panels will be assembled into a table. The table supplies the borders.

Add the words

Use your storyboard as a guide, but feel free to revise.

- **Use a new layer!!!!**

Do not start typing on any of your other layers. I find I need to type the words several times to get them to fit right. When they have their own layer, this is no problem. Also, I find I need to be able to revise. As time goes by, the facts change and I need to be able to change the words without destroying background pixels. Therefore, use a separate layer just for words. Remember that you have saved copies of your originals if you accidentally type on your background. Notice that you do not have the balloons yet--they come later.

1. Create a new empty layer named Words.
2. Set foreground color to black.
3. Select the Type tool .
4. Click in the top middle of where you want the text.
5. The Text dialog box will open. Select Family = Helvetica, Face = Medium, Size = 12 points, Alignment = centered. Anti-aliased = yes. These settings are somewhat optional, but you want optimum clarity, nothing fancy, and no smaller than 12 points. Whatever settings you choose, write them down somewhere permanent, to avoid guesswork when you revise the text next year.
6. Where the blinking cursor is in the dialog box, type your text, using all caps as is the convention in cartoons. Then Click OK.
7. The text will be a floating selection. Use your arrow keys to position the text. Click anywhere in the image to paste it into the layer. If you don't like it, select Edit/Undo Deselect and then Backspace and select the Type tool again and start over. Text in Photoshop is just pixels; you cannot edit it as you would if the text were objects. This is where you are going to be glad you are working with layers; otherwise you would be embedding the text-pixels in your image.

- **Be brief**

You may have to condense the words several times to get them to fit into the panel. Simplify!

- **Be legible**


Check to see that you are using a plain font such as Helvetica in at least 12 point size, in all caps, in black. No italics!

- **Consider an alternative**

If your handwriting is great and you have a digitizing pen, you could hand-letter the words using the paintbrush in Photoshop.

Add balloon layer

- **New layer, *under* words**

Use a separate layer for the balloons--you will want to redraw them without having to redo your words. In Photoshop, on the Layers palette, click the new layer icon  and name the layer Balloon, then drag the Balloon layer icon between the Words layer and the People layer. Doing this will put the balloon layer behind the words. Use white paint, with a big hard brush and paint in solid balloons around (behind) the words. (If the Brushes palette is not already showing, choose Window/Palettes/Show Brushes.) To make the triangle to the mouth, hold down Alt (or Option) while clicking a triangle with the lasso. Then choose Edit/Fill/ to fill with white. To make a black line around the balloon, change the foreground color to black and then hold down Control + Alt (or Option) while typing the letter T on the keyboard to select the balloons (this selects everything on the layer). With the balloons selected, choose Edit/Stroke and in the dialog box select one pixel, inside. Then do Control D (or Select/None) to undo the balloon selection. If you need to revise the balloons, select the balloon layer, select the balloons with the marquee tool, delete, and brush them in again.



Composite Alternative

The WebToons in existence so far are composed of separate images made into separate imagemaps, arranged in tables. It could be argued that working with one big composite of the images would be easier. If you want to try that, composite your images at this point. Copy and paste your layered images into one big image with appropriate borders around the images. Save the layered composite for future revisions of the WebToon. I would appreciate knowing how this works out for you: walatka@nas.nasa.gov

Adobe Photoshop is a registered trademark of Adobe Systems
Incorporated

Chapter Four

Saving Your Files

Save the Layers in Your Files!

- You **will** need your layers
- You **will** make changes

One of the great things about the web is the ease with which you can update your documents. The need for revision arises. To facilitate future changes, you will want to be sure and retain the layers in your images. In Photoshop version 3.x and higher, this usually is not a problem; just be careful not to make the mistake of flattening your image, or merging layers and then saving the file to the same filename you used for the layered image. Make a point of saving the layered files and saving backups to another machine or removable media.

Export GIF

- Photoshop/File/Export/Gif89a
CompuServe GIF is a graphic image format that works well for imagemaps and is readable on all graphics-capable web browsers. Interlaced GIF gives the user something to look at while the file is downloading. Alternately, if image quality were a big issue, you could use JPEG.

Photoshop version 3.0 and higher offers a safe and convenient method for saving GIF files.

1. Use File/Export/Gif89a. This works well and leaves your layered file unaltered. Gif89a gives you the option to interlace your gif file, which makes it appear to load faster because the user can see the gist of the image as it loads. Leave Interlace clicked on.
2. Sometimes, mysteriously, Gif89a is not available (for example, with grayscale). You can use File/Save a copy/CompuServe GIF. This method also protects your layered files, but your files will be bigger than Gif89a files and interlace is not offered.

If you are not familiar with the gif file format, look it up in the Glossary at

<http://science.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-96-002/glossary.html>

Chapter Five

Imagemaps, Tables and the Future of WebToons

WebToons Are Imagemaps

See our introduction to imagemaps in Chapter One if you are unclear on the basic idea of imagemaps.

WebToons are imagemaps arranged in tables. Each image (panel) is a separate imagemap. A typical WebToon has from five to nine imagemaps, arranged in a table to fill one web browser screen.

The Easy Way

If you are using a visual web page layout tool (such as Dreamweaver), you can make imagemaps by dragging, dropping and clicking. Then mouse some more to put the imagemaps into a table. Use your storyboard as a guide for the table.

Tables are optional; you could just insert a list of imagemaps into your HTML and let the user's browser arrange them according to space available.

Also, you may find it easier to composite all your images into one big imagemap. See information at Composite near the end of Chapter Three. Experience so far indicates that using separate imagemaps facilitates revisions, but this is debatable.

The Hard Code Way

If you like to write your own HTML, read this section. Otherwise, skip ahead to Sound, Motion and Depth.

Essentially there are two types of imagemaps: client-side and server-side. Client-side imagemaps are the newer and preferred technology. The instructions below are for client-side imagemaps. Note: in changing my old server-side imagemap files to client-side files, I found that I needed to move the "default" line to the bottom of the map list, in order for the imagemaps to work on both major brands of browsers.

What You Need to Make an Imagemap

Once you have created your WebToon images you are ready to turn each of your images into an imagemap. Remember that in a WebToon, each panel (image) is a separate imagemap.

1. Subdivide the image into clickable regions--in WebToons, the word balloons are the clickable regions. I usually use each panel to represent one chapter, and each balloon in the panel represents one section within the

chapter.

2. For each region (word balloon) you will need the following:
 1. The URL to the section of the document the region represents
 2. The region coordinates.

The URL

Each clickable region of your WebToon image will link to a specific URL (Uniform Resource Locator, web address). If your HTML files are within the same directory as the WebToons, you only have to use relative URLs, not absolute path URLs. For example, if all your files are in the `http://science.nas.nasa.gov/~walatka/WebToons` directory or folder, your URL for Chapter Three could be just `chapter3.html` instead of the full pathname.

<A NAME> Anchor

In order to link to a section within your HTML document, you will need to edit your HTML document and add anchor tags, with NAME attributes, at the top of the section. For example, ` ` That is, your imagemap can link to a particular section within the file if the section has an `<a name>` tag and closing `` tag, usually on a line by themselves just above the title of the section. In the URL in your imagemap, use the value (for example, `section2`) of the A NAME anchor, preceded by a pound (#) sign. Like this:

```
href="chapter3.html#section2"
```

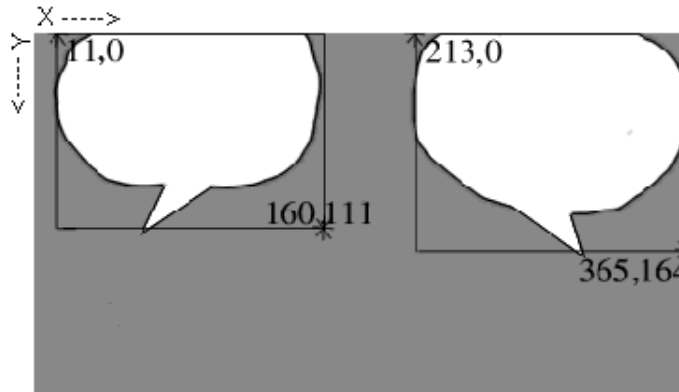
Region coordinates

You need to get the pixel coordinates of your clickable regions (the word balloon). Your image must be in its final dimensions. I use the UNIX utility `xv` or Photoshop to measure image coordinates. It is important that your clickable regions do not overlap or share coordinates. You can find automatic mapping programs on the web; see the References section for some suggestions.

Image coordinates are two-dimensional, measured in pixels. The origin (zero-zero) is at the upper-left corner of the image, with X increasing from left to right, and Y increasing from top to bottom.

The diagram below may clarify. I have sketched in temporary rectangles to roughly approximate the balloons. (The imagemap creator has the option of using rectangles, circles, or polygons. For now, let's use rectangles for simplicity, and because some browsers do not yet recognize the other shapes).

Rectangles are defined by their upper-left and lower-right coordinates. In the figure below, the upper-left corner of the first rectangle is at `x=11` and `y=0`. That is, the upper-left corner of the clickable rectangle is 11 pixels over and zero pixels down from the upper left-hand corner of the image.



The lower-right corner of the clickable rectangle is located 160 pixels over (x-direction) and 111 pixels down from the upper left-hand corner of the image.

Balloon Layer, Image for Chapter 3

The x,y coordinates are separated by commas, and pairs of coordinates are separated by commas. The coordinates for the first rectangle look like this:
 coords="11,0,160,111"

Do not overlap your coordinates. If one rectangle is 0,0,200,75 then the one below it would start at 0,76 not 0,75.

Details: I determine the coordinates by looking at the image in Photoshop, and setting the cross-hair section of the Show Info palette to pixels. [If the Info palette is not showing, choose Window/Palettes/Show_Info, then click-and-hold on the cross in the x-y box at the bottom of the Info palette and choose "Pixels."] I draw (or imagine) temporary rectangles on a new layer, and put my cursor in the upper-left corner of the rectangle. Then I note the pixel coordinates in the cross-hair section of the Info palette. Next I note the coordinates of the lower-right corner. This process is repeated for each clickable region.

When you have your A NAME tags, and your region coordinates and corresponding URLs, you are ready to write the HTML code for the imagemaps. (Mapping programs probably help you with this.)

Imagemap HTML Code

To write HTML for an imagemap, first you declare the imagemap and then you have your image use the map. There are three elements of an imagemap (details follow):

1. The map name
2. The list of coordinates for clickable areas (regions) and their corresponding target URLs
3. The map image, in GIF format

Here is a sample

```
<map name="3toon.map"> <area shape="rect" coords="11,0,160,111"
href="chapter3.html">
<area shape="rect" coords="213,0,365,164"
href="chapter3.html#section2">
<area shape="default" href="chapter3.html">
</map> 
```

```
USEMAP="#toon3.map" ALT="toon3.map">
```

Note the # sign before the map name.

The map name

Give the map a name. This is done via the map tag's NAME attribute, for example:

```
<map name="3toon.map">
```

The region list

For *each* of the imagemap clickable regions, create an area tag:

```
<area shape="option" coords="coords_list" URL>
```

1. The area tag has a shape attribute. Its value may be:
 - *rect* for rectangular areas
 - *circle* for circular areas
 - *poly* for polygonal areas
 - *default* for background areas--in case the user clicks inside the imagemap but outside of the defined regions. ****Always put the default line last in your list of regions.****

Some current browsers recognize only the rectangular shape and default.
2. The area tag has a coords attribute. See the "Region Coordinates" section above for instructions on getting the pixel coordinates. For rects, give the upper left and lower right coordinates in pixels. That is, in pixels, how far away is the upper left corner of the region from the upper left corner (origin) of the image? And how far from the origin is the lower-right corner? For circles, give the x and y for the center point and the radius in pixels. For polygons, the format is x comma y comma for each point of the polygon, going all the way around the polygon.
3. href -- give the URL you want to link to, in standard href format; for a file in the same directory, just give the filename. For a specific section within the file, use the filename, a pound (#) sign, and the section's <a name> tag (see the "The URL" section above).

The code for a polygon clickable region looks like this

```
<area shape="poly"
coords="79,51,145,53,146,93,133,95,61,87"
href="myfile2.html#header6">
```

Quiz

In the diagram in the previous section, if the first balloon were meant to link to the top of Chapter 3, the imagemap HTML code for the first balloon would look like this:

```
<area shape="rect" coords="11,0,160,111"
```

```
href="chapter3.html">
```

See if you can figure out the code for linking the second balloon to chapter3.html, section2. The answer is in Appendix B.

The image and the end of the imagemap

In addition to giving the map name and a list of regions, you will need to call the image and close the map. See the example below.

```
<map name="3toon.map"> <area shape="rect" coords="11,0,160,111"
href="chapter3.html">
<area shape="rect" coords="213,0,365,164"
href="chapter3.html#section2">
<area shape="default" href="chapter3.html">
</map> 
```

Note the # sign before the map name.

Trouble shooting: If your map does not seem to work,

1. Double-check the punctuation of each set of coordinates:
x comma y comma x comma y comma.
2. Check that your regions do not overlap.
3. Check that your default line is last in your area list.

Putting the Imagemaps Together in Tables

WebToons put imagemaps into tables, because you can get nice borders and good alignment of panels. This section reveals how to create the table of imagemaps via a set of progressively more detailed examples.

Alternative: Instead of using a table, you could just put the imagemaps into your HTML, one after another, and let the user's browser arrange them according to space available. HTML is designed to give design authority to the browser, not the page creator.

For introductory table information, see the Table Section in WebPrep at science.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-96-002/WebPrepChp4.html#HDR. Most HTML editing programs will make tables for you--or you could borrow the sample code in the Appendix. If you want to know how to hard-code a table, read on.

Start with a hard copy of your storyboard--your layout plan for your WebToon. Sketch in (or merely notice) the layout of the rows. The example shown has three rows.

Top	Row
Middle Row	
Bottom	Row

This means the HTML document for your WebToon page will have a table with three rows. This first example is just the framework of the table rows; we will fill in the cells later.

```
<!-- ***TABLE of IMAGEMAPS for WEBTOONS*** -->
<table border width="414" >
<tr> <!-- Top Row -->

</tr> <!-- End Top Row-->
<tr> <!-- Middle Row = all one toon-->

</tr> <!-- End Middle Row -->
<tr> <!-- Bottom Row -->

</tr> <!-- End Bottom Row -->
</table>
```

When you have your rows coded, as above, you are ready to add your table data cells. Notice in the sketch to the right that this layout has two cells in the top row, one cell in the middle, and two cells in the bottom row. Also note that one cell in the middle row spans two columns.

cell 1	cell 2
cell 3	
cell 4	cell 5

The cell sketch

Here is the same table code as above but with the cell code added. Notice that each cell is delimited by a TD tag.

```
<!-- ***TABLE of IMAGEMAPS for WEBTOONS***-->
<table border width="414" >
<tr> <!-- Top Row -->
    <td> <!-- ***Toon 1 = cell 1*** -->

    </td>
    <td> <!-- ***Toon 2 = cell 2*** -->

    </td>
</tr> <!-- End Top Row-->

<tr> <!-- Middle Row = all one toon--just one cell -->
    <td colspan="2" > <!-- ***Toon 3 = cell 3*** -->

    </td>
</tr> <!-- End Middle Row -->

<tr> <!-- Bottom Row -->
    <td> <!-- ***Toon 4 = cell 4*** -->

    </td>
    <td> <!-- ***Toon 5 = cell 5*** -->

    </td>
```

```

</tr> <!-- End Bottom Row-->
</table>

```

The indents are for clarity, not required.

Take a look at the HTML above and get a feel for the structure of the table. When you understand how the rows and cells are arranged, you are ready to plug in your imagemap code, as follows.

```

<!-- ---COMPLETE TABLE of IMAGEMAPS for WEBTOONS--- -->
<table border width="414" >
<tr> <!-- Top Row -->
  <td> <!-- ***Toon 1*** -->
    <map name="1toon.map">
      <area shape="rect" coords="0,0,200,68" href="index.html">
      <area shape="rect" coords="0,69,200,130" href="chapter1.html">
      <area shape="rect" coords="0,131,200,150" href="../home.html">
    </map>
    
  </td>
  <td> <!-- ***Toon 2*** -->
    <map name="2toon.map">
      <area shape="rect" coords="0,0,200,68" href="chapter2.html">
      <area shape="rect" coords="0,69,200,150" href="chapter2.html#photo">
    </map>
    
  </td>
</tr> <!-- End Top Row -->

<tr> <!-- Middle Row = all one toon--just one cell -->
  <td colspan="2" > <!-- ***Toon 3*** -->
    <map name="3toon.map">
      <area shape="rect" coords="0,0,107,88" href="chapter3.html#getin">
      <area shape="rect" coords="108,0,182,107" href="chapter3.html#pick">
      <area shape="rect" coords="183,0,305,63" href="chapter3.html#enhance">
      <area shape="rect" coords="306,0,410,79" href="chapter3.html#words">
      <area shape="rect" coords="87,64,308,138" href="chapter3.html#balloon">
      <area shape="default" href="chapter3.html">
    </map>
    
  </td>
</tr> <!-- End Middle Row -->

<tr> <!-- Bottom Row -->
  <td> <!-- ***Toon 4*** -->
    <!-- in my sample WebToon this one is just a linked image, not an imagemap -->
    <a href="chapter4.html"> </a>
  </td>

```

```

<td> <!-- ***Toon 5*** -->
<map name="5toon.map">
<area shape="rect" coords="0,0,200,64" href="chapter5.html">
<area shape="poly" coords="70,70,150,70,170,90,170,130,
150,130,70,140,50,130,50,90,70,70" href="chapter5.html#html">
<area shape="default" href="chapter5.html">
</map>

</td>
</tr> <!-- End Bottom Row -->
</table>

```

Summary of Imagemaps-in-Table HTML

1. (Remember that tables are optional; you could let the user's browser arrange your imagemaps.)
2. Make an HTML table that corresponds to the WebToon storyboard. For each horizontal row of the storyboard, there should be a corresponding row (TR) in the table. For each panel (image) in the row, there should be a corresponding table cell (TD).
3. Place the HTML code for one imagemap within each table cell.
 - Open the cell <TD>.
 - Name the map: <map name="1toon.map">
 - Add the area tags: shape, coords, and href for each clickable area within the toon image (each word balloon) plus an area called "default" linking to the top of the chapter, for the regions outside of your defined areas (default is optional).
 - Close the map: </map>
 - Call the image, giving width, height, alt, and USEMAP="toon-name.map".
 - Close the </TD>.
4. When you are at the end of the row, close the TR.
5. When you are at the end of the Toon, close the table.

What About Sound and Motion and Depth?

This is being written in 1998. I think that the current average user's bandwidth calls for quiet, static, flat WebToon images. Plain images are neither too much nor too little to download.

In the very near future, the average user's bandwidth, coupled with multimedia delivery technology, will allow WebToons to add sound; the WebToons could talk out loud. Also, the WebToons could move; there is no inherent reason for them to be static images. Animation is a cornerstone of modern cartoons. And then we could add depth; why not have 3D WebToons?

For the time being, quiet, static, flat WebToon images get the job done.

Summary of How To Make WebToons

- Make a storyboard summarizing your document.
- Create digital images of people talking.
- Copy, save, pick, crop the images.
- Enhance the images in Photoshop or other image processing software.
- Add words on new layers and balloons on new layers.
- Save layered files and GIF files.
- Create an imagemap for each GIF image.
- Assemble the imagemaps into an HTML table (optional).
- Place the table (or series of imagemaps) into an HTML document that will serve as your Table of Contents. Optional: also provide a text-only Table of Contents, on another page, for those who prefer reading plain text.
- Alternately, you could composite your Photoshop images into one big image, and make one imagemap from that.

Summary of Why to Make a WebToon

A WebToon adds the human touch to your long document on the web. Images of people make your document more approachable. Also, the creation of a WebToon forces you to organize your document such that you can condense it into logical words that fit into word balloons. Thus, WebToons help humanize and organize your long documents. WebToons enhance the usability of your web site.

If you have any questions, comments or recommendations, please write to me: walatka@nas.nasa.gov. If you succeed in making a WebToon, I would be interested in hearing about it. Good luck!

Acknowledgments

I am indebted to Sandy Johan, Val Watson and Sharon Marcacci for excellent suggestions, and to Sam Uselton for guidance, and to Randolph Kaemmerer for help with the manuscript. IHiP provided good imagemap information on the web.

I thank all the readers of this document, especially those who try to make a WebToon.

Annotated References

1. An outline of this paper was published in: Walatka, Pamela P., "WebToons: A Method for Organizing and Humanizing Web Documents," Visual Proceedings, The Art and Interdisciplinary Programs of SIGGRAPH96, Computer Graphics Annual Conference Series 1996, Association for Computing Machinery, New York, ISBN 0-89791-784-7, ACM Order No. 428961, p. 156.
2. IHiP, at <http://www.ihip.com/cside.html> gives good information and links to mapping tools.
3. www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Imagemaps/ has links to imagemap information, including imagemap editors.
4. science.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-96-002/WebPrepChp4.html is part of my guide to putting scientific reports on the web. Chapter Four is about imagemaps; the Glossary explains many of the terms used in this document.
5. The NAS Webweaver's page science.nas.nasa.gov/Services/Education/Resources/webweavers.html links to good resources for web workers, including guides to HTML.
6. Rick Darnell, et al., HTML 4, Sams.net Publishing, 1998, is a thick, complete book on HTML, including tables and imagemaps. See page 222 for imagemap tools.

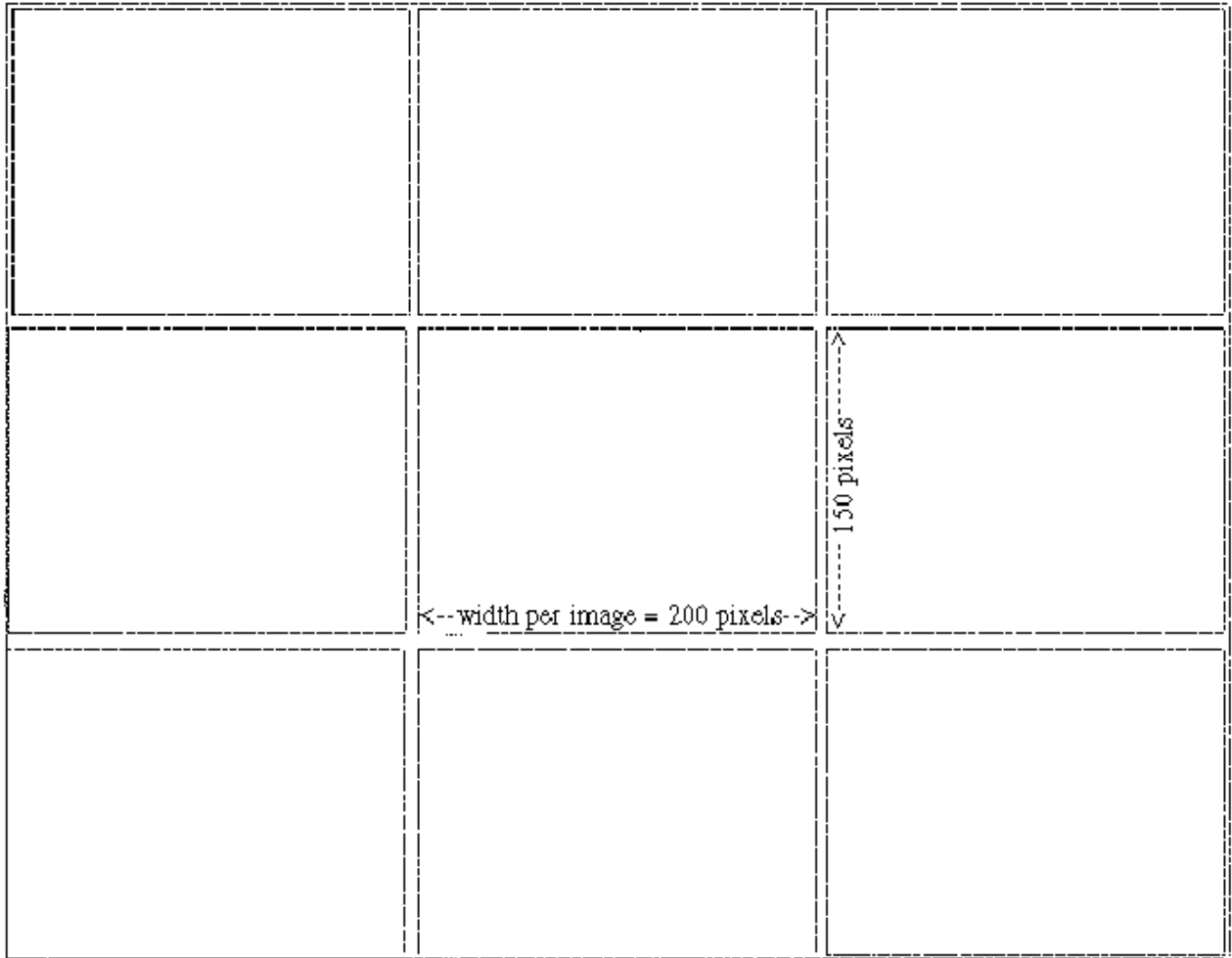
This is <http://science.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-98-014>

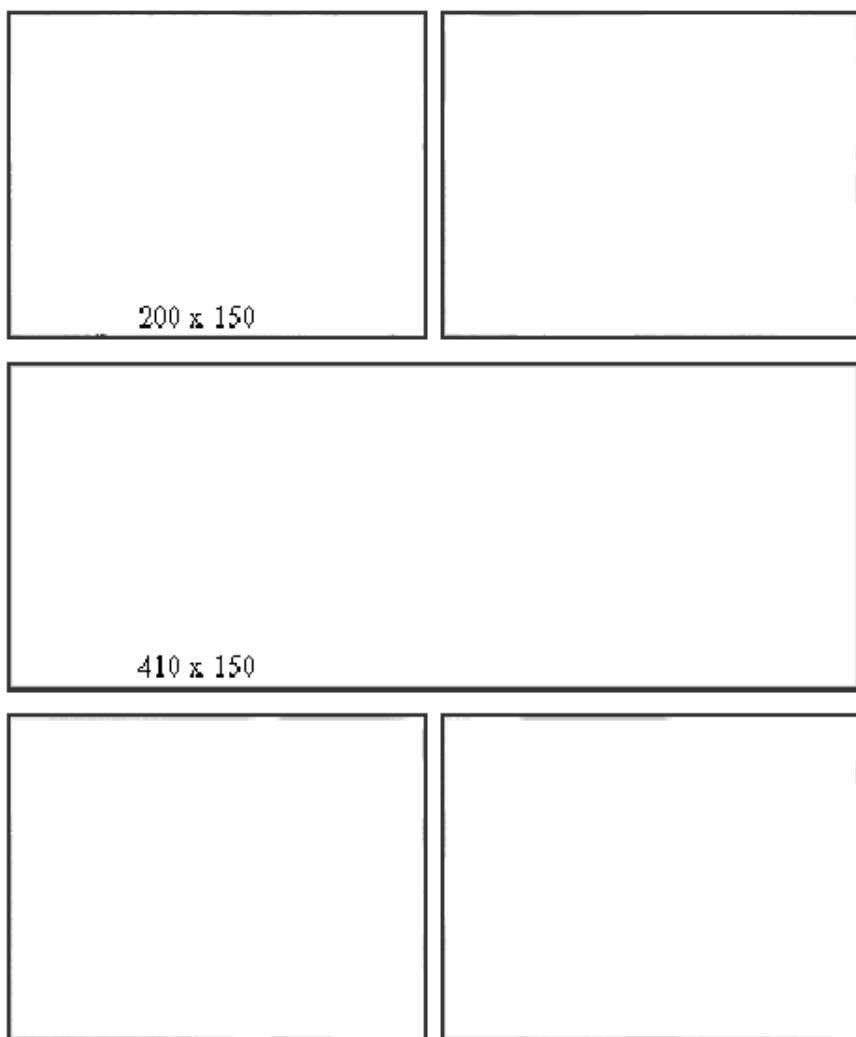


Updated: February 18, 1999
WebWork: Pam Walatka
NASA Responsible official:
Lisa Reid

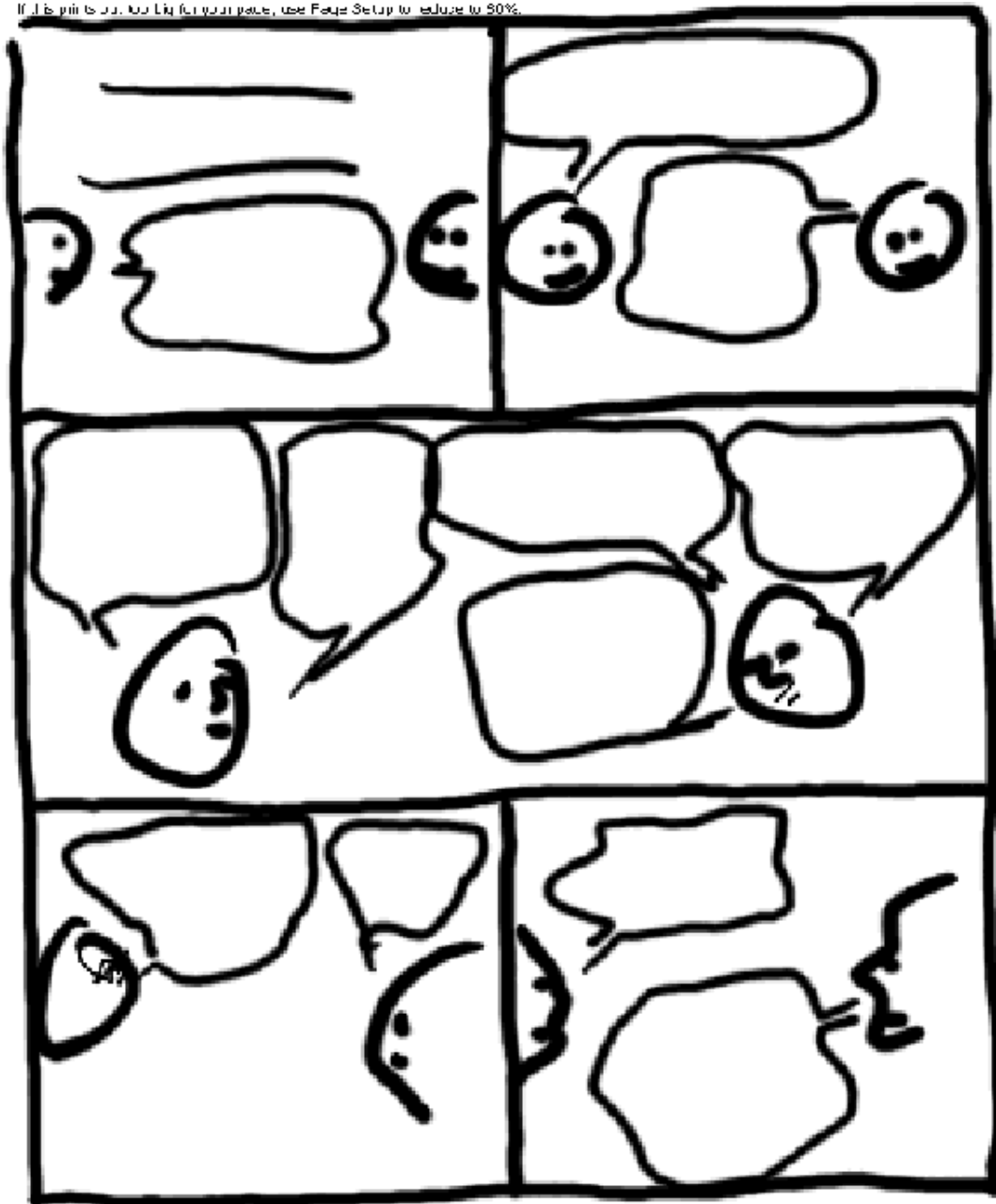
Appendix A

Sample Storyboards





If it is still too big for your page, use Page Setup to reduce to 80%.



Appendix B

Map Coordinates Quiz Answer

```
<area shape="rect" coords="213,0,365,164" href="chapter3.html#section2">
```

Appendix C

Sample HTML for Imagemap Tables for WebToons

```
<!-- ***TABLE of IMAGEMAPS for WEBTOONS***-->
<table border width="414" >
<tr> <!-- Top Row-->
<td>
<!-- ***Toon 1***-->
  <map name="1toon.map">
    <area shape="rect"      coords="0,0,200,68"      href="index.html">
    <area shape="rect"      coords="0,69,200,130"     href="chapter1.html">
    <area shape="rect"      coords="0,131,200,150"     href=" ../home.html">
  </map>
  
</td>
<td>
<!-- ***Toon 2***-->
  <map name="2toon.map">
    <area shape="rect"      coords="0,0,200,68"      href="chapter2.html">
    <area shape="rect"      coords="0,69,200,150"     href="chapter2.html#photo">
  </map>
  
</td>
</tr> <!-- End Top Row-->

<tr> <!-- Middle Row = all one toon-->
<td colspan="2">
<!-- ***Toon 3***-->
  <map name="3toon.map">
    <area shape="rect"      coords="0,0,107,88"      href="chapter3.html#getin">
    <area shape="rect"      coords="108,0,182,107"    href="chapter3.html#pick">
    <area shape="rect"      coords="183,0,305,63"     href="chapter3.html#enhance">
    <area shape="rect"      coords="306,0,410,79"     href="chapter3.html#words">
    <area shape="rect"      coords="87,64,308,138"    href="chapter3.html#balloon">
    <area shape="default"   href="chapter3.html">
  </map>
  
</td>
</tr> <!-- End Middle Row-->

<tr> <!-- Bottom Row-->
<td>
<!-- ***Toon 4***-->
  <!-- this one is just a linked image, not an imagemap-->
  <a href="chapter4.html"> </a>
</td>
<td>
<!-- ***Toon 5***-->
  <map name="5toon.map">
    <area shape="rect"      coords="0,0,200,64"      href="chapter5.html">
    <area shape="poly"      coords="70,70,150,70,170,90,170,130,
      150,130,70,140,50,130,50,90,70,70" href="chapter5.html#html">
  </map>
</td>
```

```

        <area shape="default"                                href="chapter5.html">
    </map>
    
</td>
</tr> <!-- End Bottom Row-->
</table>
<!-- End WEBTOON-->

```